

Continuous Integration and CruiseControl.NET

Owen Rogers

orogers@thoughtworks.c
om

Mike Roberts

mroberts@thoughtworks.com

Overview

- Introduction to Continuous Integration
- Introduction to CruiseControl.NET
- Demo
- CruiseControl.NET Architecture
- Extending CruiseControl.NET

What is Continuous Integration?

- “The **practice** of **integrating** source code **continuously**”
- What does that mean? And why should I do it?

What do you mean by integration?

- At a minimum:
 - Gather all source together
 - Compile
 - Execute comprehensive set of tests
 - Verify compilation and tests were successful

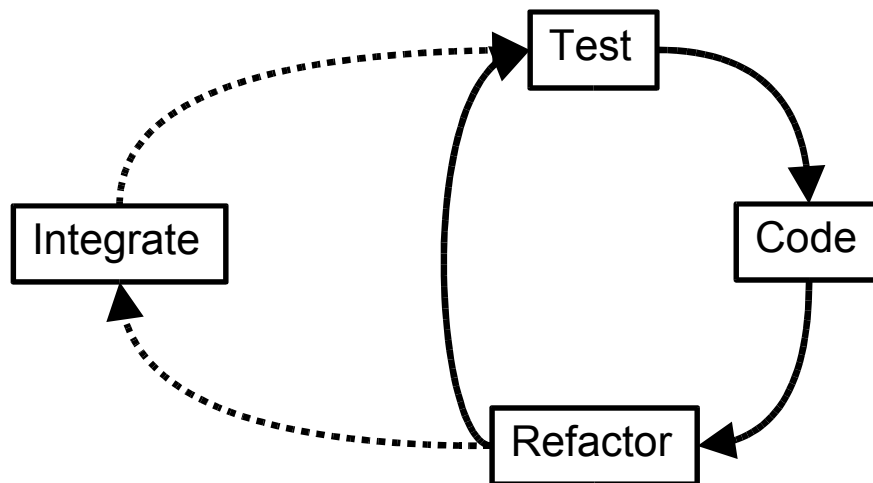
- Can include other tasks such as:
 - Rebuild the database
 - Build release distribution
 - Run code analysis and coverage tools
 - Generate documentation
 - Any scheduled task

How often is continuously?

- As frequently as possible
- More like once per hour than once per day
- Before leaving at the end of the day
 - Or throw it away

How do you practice continuous integration?

- Implement just enough, integrate, and plan what to do next
- Natural break in the development cycle
- All about taking small steps



Why practice Continuous Integration?

- Reduce integration pain
- Regular feedback
- Easier to diagnose and fix problems
- Enables concurrent development
- Minimise bottlenecks
- Code always compiles and tests run successfully
- Clean machine
- Lowers risk
- Less stress
- Creates happy developers

What do you need to practice Continuous Integration?

- Comprehensive, quick set of tests
- Single-step integration process
- Discipline to regularly integrate

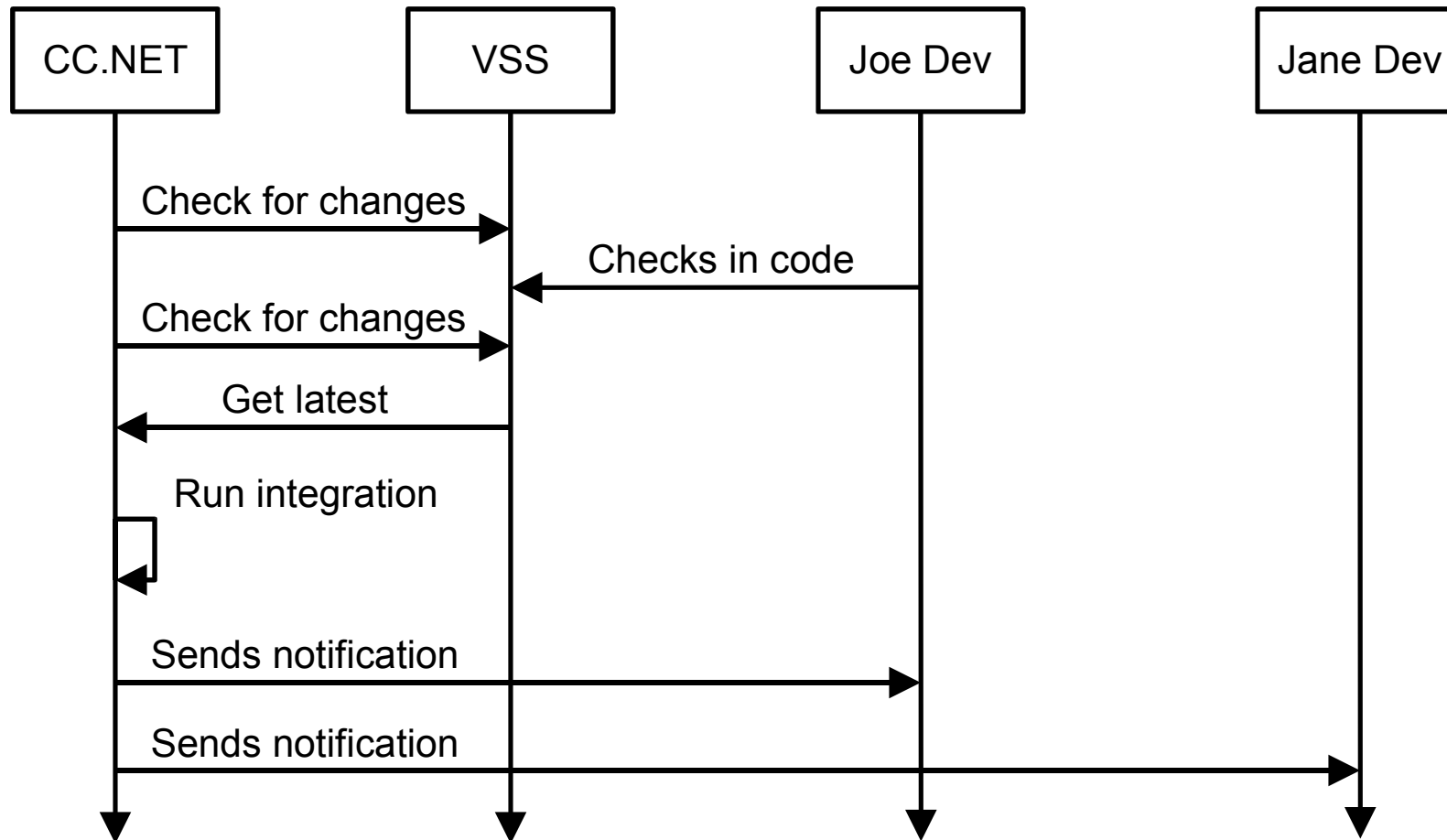
What is CruiseControl.NET?

- Automated integration server
 - Detects source changes in source control
 - Launches integration build
 - Publishes results (web page, email, net send, Yahoo)

Why use an automated integration server?

- Makes integration easy (automatic)
- Guarantees integration happens
- Rapid notification of integration results
- Identifies responsibility for fixing the build
- Encourages test automation
- Encourages agile practices

How does CruiseControl.NET work?



CruiseControl.NET Etiquette

- Fixing the build is the highest priority
- You broke it, you organise fixing it
- Do not check in on a broken build
- Check in before leaving at the end of the day
- Don't leave until the integration runs successfully
- Keep the build quick

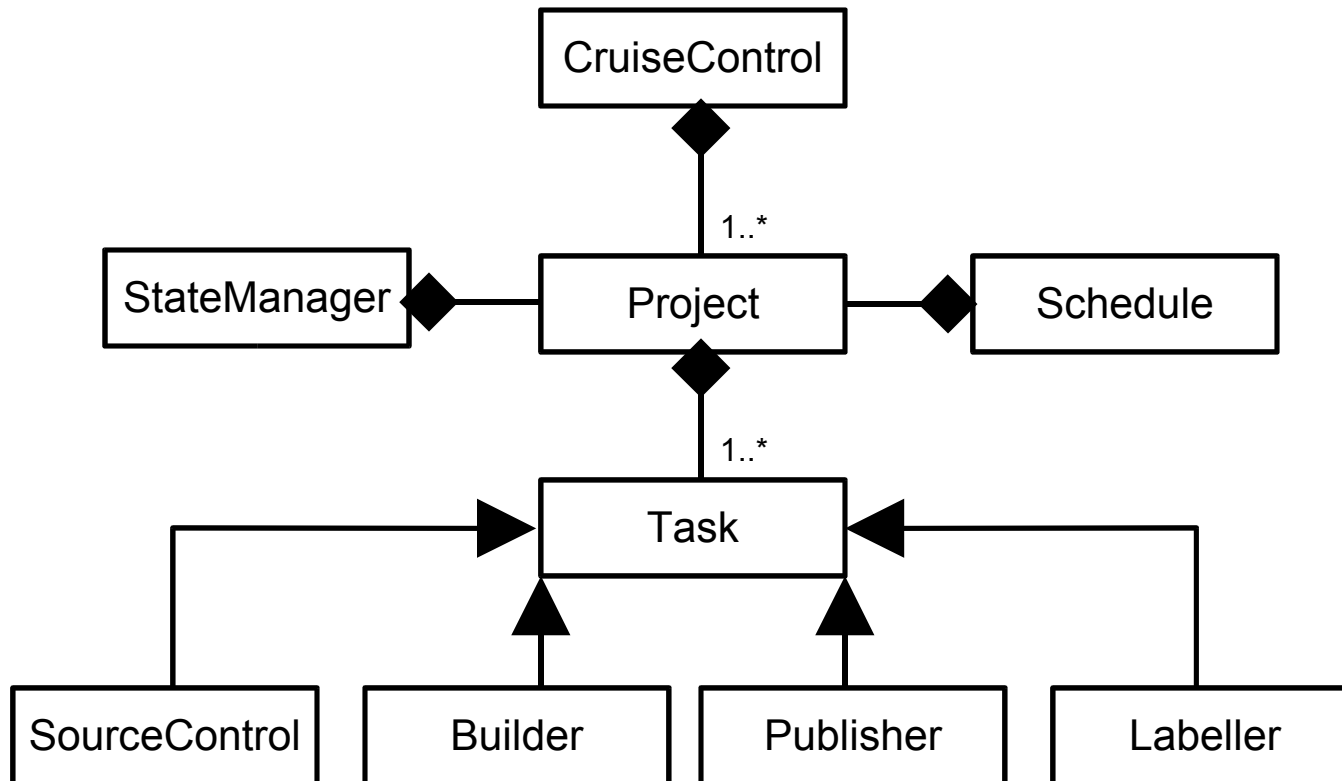
Continuous Integration != CruiseControl.NET

- Continuous Integration without CruiseControl.NET
- CruiseControl.NET without Continuous Integration

Demo

- Over to you, Mike!

CruiseControl.NET Server Architecture



How do you extend CruiseControl.NET?

- Pluggable tasks
- Pluggable projects
- Custom clients

How do you diagnose integration problems?

- Notification of broken builds
- Consult web page for integration
- Check XML log file
- Check CCNet console output (instrumentation)

Where can you get CruiseControl.NET?

- Binaries and documentation:
 - <http://ccnet.thoughtworks.com>
 - <http://continuousintegration.net>
 - <http://ccnet.sourceforge.net>
- Source:
 - <http://sourceforge.net/projects/ccnet>